Decoding the DCF77 Signal on a Raspberry Pi

René Ladan

November 2018

Overview

- Transmitter
- Reception Hardware
- Software
- Third Parties
- Closing

Transmitter → DCF77

What is DCF77 ?

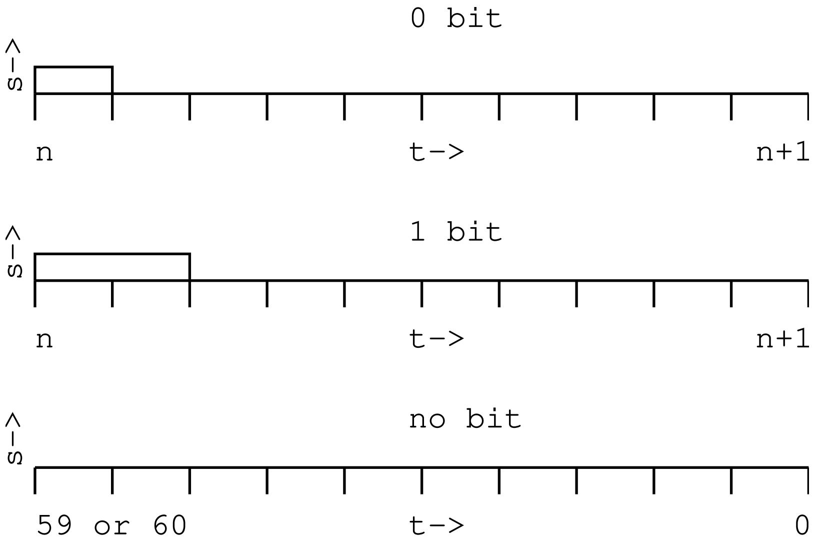
- DCF77 is a time station located near Mainflingen, Germany (50°01' N 09°00' E)
- Transmits at 77.5 kHz with 50 kW
- Operational since 1957, time/date code since 1973-07-05 [1]
- Operated by the Physikalisch-Technische Bundesanstalt

Transmitter → Others

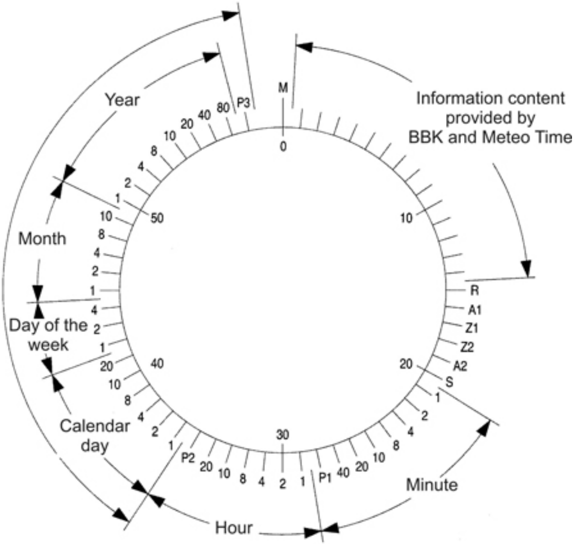
DCF77 is not the only transmitter

- "Time from NPL" (formerly MSF Rugby) in the UK at 60 kHz
- WWVB in the USA at 60 kHz
- CHU in Canada on shortwave, digital mode
- . . .

Transmitter → Modulation



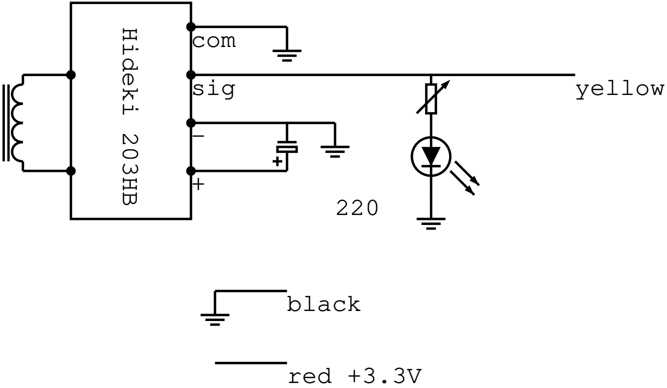
Transmitter → Date/time Format



(image from the DCF77 website [2])

Reception Hardware

Regular Raspberry Pi B with this receiver connected to the GPIO pins



Software → Overview

- Development at `https://github.com/rene0/dcf77pi`
- First commit at 2013-05-14
- Grew into a layered design
- Runs on FreeBSD and Linux
- Doxygen supported
- BSD licensed

- More advanced than my first decoder in GW-BASIC around 1990 ☺

Software → Major Versions

- 1.X → line-mode client, input from receiver and log file
- 2.X → Live reception uses curses
- 3.X → Extract library, readpin as client, some unit tests, decode alarm messages

Software → Reception Hardening

- reset realfreq if out of bounds
- reset lengths of bit 0 or bit 20 if they drift too far from the target or from each other
- only update realfreq if everything is fine
- check BCD values of date and time next to the parity bits
- detect sudden jumps in date and time
- received minutes are only valid after some sanity checks (length, bit 0/20, 17/18)

Software → Feature Evolution

- Bit detection method
- Hardening against bad weather
- Configuration file
- Analysis of received date and time
- Handling of leap second and daylight saving time
- Library interface
- Coding style

Software → Current and Future Ideas

- kqueue(2) instead of nanosleep(2) to receive pulses
- weather decoder
- UDP stream of raw radio signal → app
- "Time from NPL" should be within reception range

Third Parties → Bits for Hire

Bits 1 to 14 were used in the early days to transmit the difference between UTC and UT1, and later to extend bit 15, but are now rent to third parties

- Civil warning system
- Weather forecast (MeteoTime GmbH)

Both services use messages that span 3 minutes, so 42 bits in total

- Bits 0 and 7 are 0 → 40 bit weather message
- Bits 0 and 7 are 1 → 40 bit alarm message
- Otherwise → error

Third Parties → Users

Civil warning system

- Bundesamt für Bevölkerungsschutz und Katastrophenhilfe
- experimental only, no definite format: 4 regions, 2 checksums

Meteotime

- needs HKW 581 chip for decryption, order separately
- mainly targeted at consumer device manufacturers
- little success with email/phone regarding permission

Closing → Acknowledgements

- Udo Klein for improved bit detection algorithm [3]
- Anonymous user for the idea to set the system time
- Karl Wenzelewski for allowing to somewhat decode alarm messages
- JsBergbau for an implementation to flush the log file each minute

1. http://www.ptb.de/cms/fileadmin/internet/fachabteilungen/abteilung_4/4.4_zeit_und_frequenz/pdf/2011_PTBMitt_50a_DCF77_engl.pdf
2. https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/ag-442/dissemination-of-legal-time/dcf77/
3. http://blog.blinkenlight.net/experiments/dcf77/binary-clock
   comment 5916

Closing → Audience

Questions & Answers

Demo Time